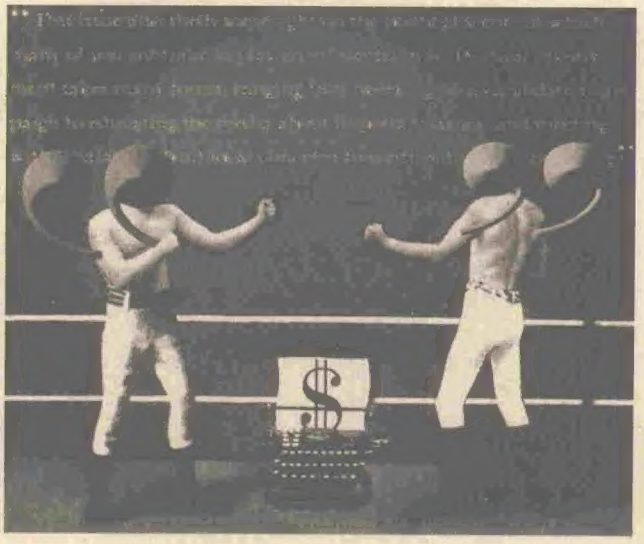
■ ไกรสร พงษ์รักษา



ต้านยูนิกซ์แครกเกอร์ ตอนที่ 1

วิธีการเบื้องต้นที่แครกเกอร์มักใช้ และการป้องกันที่ผู้ดูแลระบบจะต้องตรวจสอบว่าระบบที่ใช้งานมีช่องโหว่นั้นหรือไม่

เซิร์ฟเวอร์ที่ทำหน้าที่ให้บริการต่าง ๆ ในอินเทอร์เน็ตโดยส่วนใหญ่แล้วมักใช้ระบบปฏิบัติการยูนิกซ์ การเจาะระบบเข้ามาโดยแครกเกอร์จึงมีได้หลายเส้นทางและวิธีการตามลักษณะการให้บริการที่ยูนิกซ์มีให้ ในบทความนี้จะกล่าวถึงวิธีการเบื้องต้นที่แครกเกอร์มักใช้ และการป้องกันที่ผู้ดูแลระบบจะต้องตรวจสอบว่าระบบที่ใช้งานมีช่องโหว่นั้นหรือไม่

พอร์ตและการให้บริการ

โปรโตคอลประยุกต์ในอินเทอร์เน็ตจะกำหนดหมายเลขประจำแต่ละโปรโตคอล เพื่อแยกแยะชนิดการให้บริการโปรโตคอลปลายทาง หมายเลขประจำโปรโตคอลที่ใช้งานเป็นตัวเลขขนาด 16 บิต และเรียกว่า พอร์ต (port) หรือกล่าวอีกนัยหนึ่งแล้ว พอร์ตเป็นเสมือนแอดเดรสประจำโปรโตคอลในชั้นประยุกต์

พอร์ตหมายเลข 1 ถึง 1023 ไว้สำหรับโปรโตคอลประยุกต์มาตรฐานโดยเรียกเลขพอร์ตชุดนี้ว่า "well-known ports" ทุกสถานีหรืออุปกรณ์เครือข่ายจะใช้เลขพอร์ตมาตรฐานเพื่อให้บริการโปรโตคอลตามกำหนดเสมอ ตัวอย่างเช่น บริการ SMTP ใช้พอร์ต 25 บริการ FTP ใช้พอร์ต 21 หรือบริการ TFTP ใช้ UDP พอร์ต 69 เป็นต้น

เมื่อเรียกใช้บริการใด ๆ ซอฟต์แวร์ไคลเอนต์จะติดต่อไปยังเซิร์ฟเวอร์ตามหมายเลขพอร์ตนั้นโดยอัตโนมัติ แต่ทว่าโปรแกรมประยุกต์อย่างเช่น telnet สามารถนำมาใช้เพื่อติดต่อกับพอร์ตที่ต้องการ และเปิดโอกาสให้ทั้งผู้ใช้ทั่วไปและแครกเกอร์สามารถใช้บริการโปรแกรมประยุกต์นั้นแบบโต้ตอบได้ ตัวอย่างเช่น เมื่อต้องการขอดูหน้าจอแรกของเว็บเซิร์ฟเวอร์ใดก็ตาม สามารถใช้ telnet ไปยังพอร์ตหมายเลข 80 ของเซิร์ฟเวอร์นั้น เช่น

```
$ telnet sample.host.domain 80
Trying 10.10.2.1...
Connected to sample.host.domain
Escape character is '^]'.                    คำสั่งขอข้อมูล
GET / HTTP1.0
HTTP/1.1 200 OK
Date: Sun, 28 Mar 1999 04:02:53 GMT
Server: Apache/1.3.0                                    เฮดเดอร์ข้อมูล
Last-Modified: Sat, 27 Mar 1999 15:00:26 GMT
ETag: "165b-306d-36fcf28a"
Accept-Ranges: bytes
Content-Length: 12397
Connection: close
Content-Type: text/html                  ตั้งแต่บรรทัดนี้ไปแสดงถึงข้อมูลไฮเปอร์เท็กซ์ที่ได้

<HTML>
<HEAD>
<TITLE> Welcome to sample.host.domain home page</TITLE>
<STYLE TYPE="text/css">
A {COLOR: #0000FF; TEXT-DECORATION: none}
A:hover {COLOR: #FF0000}
</STYLE>
</HEAD>
<BODY BGCOLOR="#C0FFC0" TOPMARGIN=0>
<!— Begin header ————————————————————————>
<TABLE WIDTH=100% BORDER=0 CELLPADDING=0 CELLSPACING=0>
<TR BGCOLOR=GREEN><TD ALIGN=CENTER COLSPAN=2><IMG
:
:

</BODY>
</HTML>
Connection closed by foreign host.
$
```

ปิดพอร์ตที่ไม่จำเป็น

ระบบปฏิบัติการยูนิกซ์กำหนดเลขพอร์ตที่ให้บริการไว้ในแฟ้ม /etc/inetd.conf และเมื่อพิจารณาโดยละเอียดแล้วจะพบว่ามีบริการหลายบริการที่ควรยกเลิก เนื่องจากอาจเป็นช่องทางให้แครกเกอร์เข้ามาโจมตีจนกระทั่งระบบไม่สามารถทำงานได้ พอร์ตเหล่านี้ได้แก่ ยูดีพีพอร์ต ซึ่งใช้ตรวจสอบระบบ เช่น chargen daytime หรือ echo เป็นต้น

ตัวอย่างการทำงานของพอร์ตเหล่านี้เช่น เมื่อ telnet ไปยังเซิร์ฟเวอร์ปลายทางผ่านพอร์ต chargen เซิร์ฟเวอร์จะส่งรหัสอักขระประจำระบบกลับมาอย่างต่อเนื่อง ซึ่งไม่ได้ใช้ประโยชน์ในการบริการพื้นฐานใด ๆ แต่กลับสร้างทั้งภาระงานให้กับเครื่องและเครือข่ายโดยไม่จำเป็น ดังนั้นผู้ดูแลระบบต้องปิดบริการเหล่านี้ให้หมด ดังตัวอย่างต่อไปนี้ (สัญลักษณ์ # ข้างหน้า หมายถึง ปิดบริการนั้น)

```
#echo          dgram  udp   wait   root   internal
#discard       stream tcp   nowait root   internal
#discard       dgram  udp   wait   root   internal
#daytime       stream tcp   nowait root   internal
#daytime       dgram  udp   wait   root   internal
#chargen       stream tcp   nowait root   internal
#chargen       dgram  udp   wait   root   internal
ftp    stream tcp   nowait root   /usr/sbin/tcpd in.ftpd -l -a
telnet stream tcp   nowait root   /usr/sbin/tcpd in.telnetd
#gopher stream tcp   nowait root   /usr/sbin/tcpd gn
smtp   stream tcp   nowait root   /usr/bin/smtpd smtpd
#nntp   stream tcp   nowait root   /usr/sbin/tcpd in.nntpd
#shell  stream tcp   nowait root   /usr/sbin/tcpd in.rshd
#login  stream tcp   nowait root   /usr/sbin/tcpd in.rlogind
#exec   stream tcp   nowait root   /usr/sbin/tcpd in.rexecd
#talk   dgram  udp   wait   root   /usr/sbin/tcpd in.talkd
#ntalk  dgram  udp   wait   root   /usr/sbin/tcpd in.ntalkd
#dtalk  stream tcp   waut   nobody /usr/sbin/tcpd in.dtalkd
#pop-2  stream tcp   nowait root   /usr/sbin/tcpd ipop2d
#pop-3  stream tcp   nowait root   /usr/sbin/tcpd ipop3d
#imap   stream tcp   nowait root   /usr/sbin/tcpd imapd
#tftp   dgram  udp   wait   root   /usr/sbin/tcpd in.tftpd
#bootps dgram  udp   wait   root   /usr/sbin/tcpd bootpd
#finger stream tcp   nowait root   /usr/sbin/tcpd in.fingerd
#cfinger stream tcp  nowait root   /usr/sbin/tcpd in.cfingerd
#systat stream tcp   nowait guest  /usr/sbin/tcpd /bin/ps -
auwwx
netstat stream tcp  nowait guest  /usr/sbin/tcpd /bin/netstat
time   stream tcp   nowait nobody /usr/sbin/tcpd in.timed
time   dgram  udp   wait   nobody /usr/sbin/tcpd in.timed
auth   stream tcp   nowait nobody /usr/sbin/in.identd
in.identd -l -e o
```

UNIX

ตรวจค้นบัญชีและรหัสผ่าน

แครกเกอร์อาจใช้บริการพื้นฐานเพื่อเข้ามาตรวจหาบัญชีและรหัสผ่านโดยไม่ต้องมีบัญชีที่เซิร์ฟเวอร์ปลายทาง ตัวอย่างของช่องทางที่แครกเกอร์เข้ามาอาจเป็นได้ทั้งบริการ TELNET, FTP, FINGER หรือ SMTP ที่ใช้กับจดหมายอิเล็กทรอนิกส์ เป็นต้น

TELNET

telnet เป็นโปรแกรมให้บริการล็อกอินเข้าสู่ระบบ แครกเกอร์มักใช้สคริปต์หรือโปรแกรมที่พัฒนาขึ้นเพื่อเรียก telnet โดยป้อนชื่อบัญชีและรหัสผ่านอย่างอัตโนมัติซ้ำแล้วซ้ำเล่าจนอาจเข้าระบบได้ (เรียกว่า brute-force attack) วิธีป้องกันโดยตรงอาจทำได้ยากเนื่องจากโปรแกรมไม่สามารถแยกแยะว่าผู้ขอเข้าใช้เป็นตัวจริงหรือเป็นแครกเกอร์ การป้องกันจึงมักทำโดยทางอ้อมคือจัดเตรียมโปรแกรม telnet ที่มีคุณสมบัติหน่วงเวลารอหากป้อนชื่อบัญชีหรือรหัสไม่ถูกต้อง และเวลาหน่วงจะเพิ่มค่ามากขึ้นทุกครั้งที่ล็อกอินไม่สำเร็จ ส่งผลให้การเดาสุ่มไม่ประสบผล

SMTP

บริการพื้นฐานที่มีทั่วไปในทุกเซิร์ฟเวอร์ได้แก่ จดหมายอิเล็กทรอนิกส์ โดยใช้โปรโตคอล เอสเอ็มทีพี (SMTP : Simple Mail Transfer Protocol) โปรแกรมหรือเดมอนที่ให้บริการเอสเอ็มทีพีในยูนิกซ์คือ sendmail

sendmail เป็นโปรแกรมที่มีความซับซ้อนและมีประวัติความเป็นมาที่เกี่ยวโยงกับระบบรักษาความปลอดภัยมาตลอด แครกเกอร์มักใช้ sendmail เพื่อตรวจหาชนิดและรุ่นของระบบปฏิบัติการ รวมทั้งชนิดและรุ่นของโปรแกรม sendmail เองเพื่อใช้จุดบกพร่องที่มีอยู่นั้นเจาะเข้ามาสู่ระบบ ตัวอย่างต่อไปนี้แสดงการตรวจระบบโดยใช้ sendmail เป็นช่องทาง

```
$telnet sample.host.domain 25
Trying 127.0.0.1...
Connected to sample.host.domain.
Escape character is '^]'.
220 sample.host.domain ESMTP Sendmail 8.8.7/8.8.7;
Sun, 28 Mar 1999 11:08:28 +00
```

แครกเกอร์สามารถใช้คำสั่งใน sendmail เพื่อตรวจสอบหรือคาดเดาชื่อบัญชีผู้ใช้ในระบบ เช่น

```
vrfy root
250 root <root@sample.host.domain>
vrfy john
550 john... User unknown
vrfy somchai
250 somchai <somchai@sample.host.domain>
```

แครกเกอร์สามารถใช้คำสั่ง expn เพื่อตรวจว่าผู้ดูแลระบบติดตั้งคำสั่งย่อใดไว้บ้าง หากไฟล์คำสั่งนั้นเปิดให้ผู้ใช้ทั่วไปเขียนทับได้ แครกเกอร์อาจเขียนสคริปต์สั้น ๆ เพื่อแก้ไขไฟล์คำสั่งย่อนั้นให้เปลี่ยนเป็นคำสั่งใด ๆ เพื่อให้แครกเกอร์เข้าระบบได้โดยไม่ต้องใช้รหัสผ่าน

```
expn samplealias
250 <"|/script.pl>
```

อีกพารามิเตอร์หนึ่งที่เป็นช่องทางการเข้าเจาะระบบคือพารามิเตอร์ debug สำหรับตรวจการทำงานของ sendmail ผู้ดูแลระบบควรปิดการใช้งานพารามิเตอร์ตั้งแต่การติดตั้งเริ่มต้น และสามารถตรวจสอบว่าไม่สามารถใช้พารามิเตอร์ได้โดยพิมพ์คำสั่ง debug ดังตัวอย่าง

```
debug
500 Command unrecognized: "debug"
```

FINGER

คำสั่ง finger ใช้รายงานสถานะบัญชีผู้ใช้เครื่อง เมื่อใช้ finger ตรวจสอบผู้ใช้รายใด จะปรากฏข้อมูลเกี่ยวกับชื่อบัญชี ชื่อผู้ใช้ เชลล์ที่ใช้งาน ไดเรคทอรีบ้าน วันเวลาที่เข้าใช้ล่าสุด ตลอดจนข้อมูลอื่น ๆ

finger ให้บริการผ่านพอร์ต 79 เมื่อใช้ telnet เข้าสู่พอร์ตนี้และป้อนชื่อบัญชี จะปรากฏข้อมูลสถานภาพของผู้ใช้บัญชีรายนั้น ดังตัวอย่าง

```
$telnet sample.host.domain 79
root
Login: root                    Name: root
Directory: /root               Shell: /bin/bash
Last login Wed Mar 24 14:30 (ICT) on tty2
New mail received Sun Mar 28 04:10 1999 (ICT)
     Unread since Wed Mar 24 14:30 1999 (ICT)
No Plan.

somchai
Login: somchai                 Name: Somchai Jaidee
Directory: /home/users/somchai Shell: /bin/sh
Never logged in.
No mail.
No Plan.
```

UNIX

เป้าหมายหนึ่งของแครกเกอร์คือผู้ใช้บัญชีซึ่งไม่เคยล็อกอินเข้าสู่ระบบเลย ทั้งนี้เพราะหากลักลอบใช้บัญชีได้แล้วมักไม่มีผู้ใดสังเกตรู้ ผู้ดูแลระบบสามารถป้องกันจุดอ่อนนี้ได้โดยไม่อนุญาตให้ใช้ finger จากโฮสต์ภายนอก ยกเว้นการเรียกใช้จากเครื่องเดียวกัน วิธีดำเนินการอาจจัดหาโปรแกรม finger ที่ออกแบบมาเฉพาะหรือใช้ tcpwrapper กำหนดรูปแบบการให้บริการ finger ตัวอย่างต่อไปนี้แสดงถึงเซิร์ฟเวอร์ซึ่งไม่ให้บริการ finger

```
$ finger @sample.host.co.th
[sample.host.co.th]
finger: connect: Connection refused
```

FTP

การถ่ายโอนแฟ้มด้วยเอฟทีพีเป็นอีกบริการที่มีให้ใช้อย่างแพร่หลายในเซิร์ฟเวอร์ทั่วไป และเป็นอีกช่องทางหนึ่งที่แครกเกอร์มักใช้เข้ามาสุ่มตรวจหาบัญชี ทั้งนี้เพราะบางระบบอาจไม่บันทึกการเข้าใช้เครื่องผ่านเอฟทีพีโดยละเอียด ส่งผลให้ตรวจค้นได้ยากว่าการเจาะระบบนั้นเข้ามาทางใดและด้วยวิธีใด ตัวอย่างต่อไปนี้แสดงการใช้เอฟทีพีผ่านพอร์ต 21 เพื่อเข้ามาหารหัสผ่าน

```
$telnet sample.host.domain 21
Trying 10.10.2.1...
Connected to sample.host.domain.
Escape character is '^]'.
220 sample.host.domain FTP server
(Version wu-2.4.2-      academ[BETA-18](1)
user somchai
331 Password required for somchai.
pass guess
530 Login incorrect.
```

เซิร์ฟเวอร์ที่เปิดบริการเอฟทีพีจะไม่มีทางป้องกันการเข้าใช้บริการตามตัวอย่างข้างต้นได้เลย มาตรการที่ทำได้มีเพียงแต่การเปิดระบบบันทึกการใช้งานเอฟทีพีเพื่อใช้เป็นข้อมูลตรวจหาเมื่อพบว่ามีผู้พยายามเจาะระบบผ่านเอฟทีพี โดยติดตั้งเอฟทีพีใน /etc/inetd.conf ตามลักษณะต่อไปนี้

```
ftp stream tcp nowait root   /usr/sbin/tcpd in.ftpd -l -a
```

นอกจากนี้บริการ "anonymous FTP" เป็นอีกช่องทางหนึ่งที่แครกเกอร์มักนิยมใช้เพื่อเข้ามาลักลอบนำชื่อบัญชีผู้ใช้ทั้งหมดไปแครกหารหัสผ่าน ดังตัวอย่างเช่น

```
$telnet sample.host.domain 21
user anonymous
331 Guest login ok, send your complete e-mail address
as  password.
pass cracker@crack.net
230 Guest login ok, access restrictions apply.
```

วิธีป้องกันปัญหาคือต้องติดตั้งบริการนี้ให้ถูกต้องตามขั้นตอน ชื่อบัญชีของเซิร์ฟเวอร์ผู้ให้บริการเอฟทีพีจะต้องไม่มีรหัสผ่านและไม่มีเชลล์ที่ใช้งานได้ ดังตัวอย่างข้างล่างนี้จะพบว่าไม่มีการให้เชลล์กับ ftp (เป็นค่า null)

```
$ finger ftp
Login: ftp                    Name: FTP User
Directory: /home/ftp          Shell: null
Never logged in.
No mail.
No Plan.
```

ส่งท้าย

ต้านแครกเกอร์ครั้งถัดไปจะกล่าวถึงเทคนิคอื่น ๆ ที่แครกเกอร์ใช้ รวมทั้งแง่มุมที่เป็นจุดอ่อนของยูนิกซ์ซึ่งต้องระมัดระวังเป็นตอนที่สอง